A Strategy for an Enterprise-Wide Data Management Capability at the Jet Propulsion Laboratory

A White Paper

D-18550

Donald Fuhrman
PDMS System Engineer

January 31, 2000

Jet Propulsion Laboratory
California Institute of Technology

1 ABSTRACT

The Jet Propulsion Laboratory (JPL) is a Federally Funded Research and Development Center (FFRDC) operated by the California Institute of Technology that is "engaged in the quest for knowledge about our solar system, the universe, and the Earth". As such, it is the nation's lead center for the robotic exploration of space, and is responsible for a broad spectrum of space science missions and instruments. It pursues its goals through collaboration with government agencies, industrial partners, and academia. It is also developing more "faster, better, and cheaper" missions than ever before. Its growing "knowledge base" is represented by computer applications and information repositories spread across multiple geographic sites, in diverse computing environments, which represent an enormous body of knowledge.

Because of this heterogeneity in its knowledge base, JPL has recognized the need to provide its users with easier access to information. This includes the contributor as well as the casual browser. Both of these types of users perceive this body of information in an increasingly complex web of relationships. Because of the "faster, better, cheaper" paradigm of the new missions, users are being asked to perform tasks that are much more integrated than ever before. Gone are the days when a drafter just made drawings. Now, engineers and designers are building 3-dimensional models, performing analysis on the models, generating visualization "movies" as well as creating and publishing 2-dimensional drawings of their products – all on Web-enabled computers and networks.

The Laboratory is currently reengineering many engineering processes including Configuration Management, Product Data Management, and Document/Data Management. If someone were to describe a system that would embody these newly reengineered processes, it might be called the "Configuration Management, Product Data Management, Document and Data Management System" or CM/PDM/DDMS. For the purposes of this white paper, and to preserve our sanity, we will simply refer to this as the "system". This white paper will describe these new processes, the new system, and the strategy by which the Laboratory seeks to implement them.

2 ACKNOWLEDGEMENTS

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Table of Contents

3 INTRODUCTION

This document is intended to identify the strategic focus of JPL's initiative to develop an enterprise wide system to provide Configuration Management (CM), Product Data Management (PDM), and Document and Data Management (DDM). It contains high level functional requirements that are intended to guide the reader to a thorough understanding of the ultimate capabilities expected from the system. Use of the word "**shall**" or "**must**" indicates a mandatory requirement for the system. Use of the word "should" indicates a preferred capability, methodology, or attribute of the system.

This document's audience consists of sponsors and stakeholders within the JPL community and system vendors who will be asked to respond to a Request for Information (RFI). The responses to the RFI will be evaluated to discern the capabilities of commercial off the shelf products, and to help in the definition of the Request for Proposal (RFP). Candidates who respond to this RFI may be asked to participate in a demonstration program at JPL to demonstrate their product's capabilities. Hopefully (depending on the responses from the RFP), a procurement of an appropriate solution will be made to the successful vendor. This procurement would represent a commitment by JPL to a long-term relationship between JPL and the vendor.

This white paper, then should set the stage for subsequent activities that eventually will lead to the deployment of an enterprise wide CM / PDM / DDM system.

4 DEFINITIONS

- **System** - a composite of equipment, skills, and techniques capable of performing or supporting an operational role, or both. A complete system includes all equipment, related facilities, material, software, services and personnel required for its operation and support to the degree that it can be considered a self-sufficient unit in its intended operational environment. For the purposes of this paper, the "system" refers to the enterprise CM / PDM / DDM system.

- **Configuration Management** – a process employed by JPL to identify product configuration and effect orderly management of product change. It is a process that is applied throughout the entire lifecycle of the product – proposal to disposal. Configuration Management establishes and maintains consistency of a product's performance, functional and physical attributes with its requirements, design, test, build and operational environment throughout a product's life.

- **Product Data Management** – Product Data Management (PDM) is a tool that helps engineers and others manage data and the product development process. PDM systems keep track of the masses of data and information required to design, manufacture or build, and then support and maintain products.

- **Document and Data Management** – an integrated set of processes, services, tool, and tool guidelines that manages documents and data through their complete lifecycle – from authoring through reviewing, publishing, reusing, and archiving

- **Product Definition Data** – Information used to design, build and operate a spacecraft and its components or any related equipment. This includes any models, drawings, specifications, documents, etc., whose content describe the design intent of the equipment for both as-design configurations and as-built configurations.

5 BACKGROUND

JPL has "evolved" many data and document repositories over the past few decades. As a result, many of its resources are unable to support flight project users in today's "faster, better, cheaper" climate. Much of the information housed in these various repositories is redundant with other repositories and often must be re-keyed into one system after another. There are very few systems that can automatically update the contents of another system so that their contents might be kept synchronized.

The Internet technologies and methodologies are driving JPL to build a more robust information system, or knowledge management infrastructure. These new capabilities will enable JPL to meet the goals of its future missions.

6 ENVIRONMENT

JPL employs approximately 5,000 people, most of whom reside at the Oak Grove facility in Pasadena. All of these individuals should be considered potential users of the system. JPL also has facilities located in Spain, Australia and at various locations in the United States. The system must support these individuals as well as users from subcontractors or partners at other companies and universities around the world.

JPL has a heterogeneous computing environment. UNIX workstations and servers are heavily utilized along with a growing number of Windows workstations (PCs) and servers and of course, the ubiquitous Apple PC.

Additionally, JPL is doing more and more collaboration with other government agencies (NASA), subcontractors (Lockheed Martin Astronautics, Ball Aerospace, etc.), and international partners (such as CNES and ESA). These organizations also have heterogeneous systems, and such diversity is driving JPL to a standards based development environment.

Many of these standards are required for interoperability across the Internet. Web browsers such as Netscape Communicator and MS Internet Explorer have become the user's standard desktop with respect to information access. The application development environment is moving away from X-windows and MS-compatible windows to HTML, XML, and Java.

The system must comply with commonly-accepted standards, and must have a clear strategy to implement evolving standards. The supported standards should provide assurance that the system is "open". For example, Java is preferred over PERL as an Internet programming language, because it does not require platform-specific implementation.

6.1 OTHER INFORMATION REPOSITORIES

Over the years, JPL has invested significant resources in creating and maintaining dozens of information repositories. Each one has a specific purpose and set of functions germane to their tasks within the lifecycle of a project, or as in the case of the Vellum File the

Laboratory archive for documents and drawings. Listed below are examples of the kinds of repositories that must be integrated or otherwise interfaced with the enterprise system.

- **PDMS** – The Product Data Management System (formerly the Engineering Data Management System, or EDMS) exists to manage the various drawings, specifications, change instructions, and inspection documentation for hardware deliverables for the projects. It includes status accounting functions, such as Engineering Change Requirements (ECR) listings, drawing release listings, and waiver listing, as well as As-Designed Parts Lists and As-Built Parts Lists.
- **Project Libraries** – Each project has in the past created a standalone system (sometimes as simple as an NT file server) to collect and store project documentation such as management plans (budgets, schedules, etc.), design review minutes, presentations and such. Configuration management is usually an informal process in which a project librarian has the sole responsibility for maintaining the currency of the library's contents. There are currently about 50 projects at JPL, and hence 50 project libraries. Most of the newer libraries are built on Xerox's DocuShare system, while older libraries were built with proprietary database tools. Others, as mentioned above, are simply maintained within NT file servers.
- **DMIE** – The Define and Maintain Institutional Environment process created its own authoring system and document repository in order to manage JPL's policies, procedures and work instructions. This is essentially a custom designed system.
- **EPINS** – The Electronic Parts Information Network System provides the ability for electronic parts experts to support the procurement of components or assemblies for JPL projects. The EPINS database contains part parameters and reliability information needed to qualify hardware for space flight operations.
- **PFR System** – The Problem Failure Reporting system provides a database system to aid in the identification, collection, and disposition of problems or errors encountered during mission operations or during activities leading up to mission operations (for example in the testing phase).
- **Vellum File** – Historically (pre-computer-aided-design), the Vellum File was the official repository for released documents and drawings in hardcopy format (hence the name). The Vellum File has developed a proprietary data management system to help their staff keep track of information needed to perform their specific tasks. This database keeps track of document numbers and revisions, the hardcopy location, microfilm location, and security information, etc.
- **IBS** – The Oracle financial system contains cost information regarding a project's deliverables. The information contained in IBS is essential for a project's status accounting and management.
- **Organization Web Sites** – Dozens of organizations at JPL (line organizations, projects), and even individual engineers with their own Web servers are "publishing" information. Of course there is no standard for how information is presented, stored, archived, or otherwise maintained.

CAE Systems – JPL uses dozens of engineering tools to accomplish its development tasks. These tools have been identified as "strategic tools". Some of the most widely used tools are: Computervision CADDS5, Pro/Engineer, Mentor Graphics, i-Logix Statemate, DOORS (for requirements tracking), Cadence, Solidworks, Autocad. There are over 160 "strategic" tools being maintained at JPL.

7 DRIVERS

Cost and implementation schedules for information systems' development have become unwieldy in light of the complexity of information required to design, build and operate space exploration missions. More and more design and analysis are performed using CAD tools. These systems output huge amounts of data, which are used as inputs to other systems. For example, a mechanical engineer will develop a 3-D model, which in turn will be used to develop a thermal model. The results of the thermal analysis may be used to modify and optimize the mechanical model.

This iterative process is used more and more as projects strive to optimize all of their designs, to maximize performance, and minimize mass and cost. While these CAD tools can help engineers deliver better products, they are sure to deliver more data. This data in turn needs to be managed such that a user can be sure she is getting the latest official design, and also that that data is provided in a format that is useful to the end user.

8 GENERAL REQUIREMENTS

8.1 USER INTERFACE / CLIENT APPLICATIONS

As described above, the Web has become the de facto desktop. The system must provide the user interface through standard Web browsers (e.g., Netscape). The system must demand as little use of client resources as possible, with the bulk of the processing occurring at the server. Where local processing is required (on the client), Java applets should be utilized. This architecture will enable efficient maintenance and update to the application(s).

8.2 APPLICATION DEVELOPMENT AND INTEGRATION

The system must provide tools and capabilities to allow skilled users to develop their own extensions to provided object classes, user interfaces, and the features for custom applications. These tools must also support the development of interfaces between the system and other applications and/or repositories.

The tools must support the graphical modeling of data and relationships (database schema, process definitions, etc.). The tools must include a complete and documented set of Application Programming Interfaces. These should be offered in the form of methods that can be applied to any object in the database schema. The tools must support the development of platform independent user interfaces.

The system must offer a user interface development tool. This tool must be graphical in nature, and must generate either Java code, HTML, or XML pages, or both. Applications and interfaces created with this tool must both communicate with each other and be accessible through the World Wide Web.

The system must provide a command line syntax for connection from other Web-based systems, specifically a URL syntax with embedded queries.

8.3 AUTHENTICATION AND SECURITY

The system shall provide for authentication and security, such that each user shall be identified as authorized to view, create, or modify any record pertaining to her work.

Security functions should include ecryption of passwords and data, user roles and associated permissions. The system should also provide for the restriction of whole classses of information as well as individual records (metadata) or objects (drawings, documents, models, etc.).

8.4 SCALEABILITY

The system shall provide the ability for system administrators and/or project staff to scale the system up or down (i.e., based on capabilities needed by individual project customers). Scaleablity refers to – but is not limited to – the number of users, the number of records maintained by the system, the levels of product structure, the number of interfaces to other systems, or the number of processes supported within the system, etc.

8.5 STANDARDS

Where possible the system shall be developed with Internet standards or other industry standards. These include but are not limited to: Java, XML, SQL, TCP/IP, STEP, CORBA, UML, etc.

9 SUB-PROCESSES AND APPLICATION REQUIREMENTS

This section describes the high-level activities and/or functions needed to implement a complete system. Each of the domains of this enterprise system (CM, PDM, and DDM) can be subdivided into more tightly defined processes. Each sub-process in turn can be divided into units of activity, which could be described in a work instruction.

9.1 CONFIGURATION MANAGEMENT (CM)

Configuration Management has four sub-processes: Identification, Status Accounting, Change Control, and Verification and Audit. Each of these can be further decomposed, but for our purposes here, this level of detail will suffice. The system must support each of these sub-processes.

9.1.1 IDENTIFICATION

Identification is the process of determining what objects (hardware, software, documents, drawings, models, records, etc.) are to be controlled by the CM process, what naming conventions shall be used for these objects, and allocating unique identifiers (document numbers, revision codes, etc.). There are three categories of objects managed under the CM umbrella: **Configuration Items** (CI) refers to hardware and/or software deliverables; **Controlled Documents** are any document (or drawing, model, database, or other informational object) placed under configuration management; and **Records** which document an event and which does not undergo changes or revisions (e.g., an anomaly report).

9.1.2 STATUS ACCOUNTING

Status accounting refers to the process of tracking the lifecycle of each object where the phase of its development activity is recorded and reported at regular intervals (nightly in some cases). Typical status accounting reports include document status, drawing release listings, Engineering Change Request (ECR) Listings, and Indentured Parts Lists (IPL), etc. Metrics gathered about products for any given project are also valuable in the project management function. Graphical representation of this data is very valuable. For example, ECR status data should be compiled so that a project manager might have graphs showing the number of open ECRs vs. how long they've been open, total cost as a function of time, etc.

9.1.3 CHANGE CONTROL

Change Control includes the methods by which an organization facilitates and manages changes to a baselined product and all of its controlled documents. The status of proposed changes must be available as well. Definition of roles played by team members, access privileges and authority are vital components of change control. The identification of category and class of configuration items and controlled documents with respect to the signature authority of such changes is a critical element of this process as well.

9.1.4 VERIFICATION AND AUDITING

Comparing the design intent with the actual product is an important element of CM. Baseline verifications (FCAs, PCAs, etc.) and As-Designed versus As-Built difference lists are needed to understand any discrepancies in the documentation with respect to what product has actually been shipped.

9.2 PRODUCT DATA MANAGEMENT (PDM)

Product Data Management has several components and functions. For a robust PDM capability, all of the following functions are needed. A robust PDM system will encompass all of the CM functions listed above, and will automate much of the information processing and notification activities. The system must support all of the components or functions listed below.

9.2.1 ELECTRONIC VAULT

Product Definition Data is created through the use of computer-aided design (CAD) and engineering (CAE) tools (as well as through "conventional" processes, such as paper and pencil). Their products often are databases that capture the design intent of the end product. While many of these CAD systems generate 2-D representations of the product in the form of a drawing, the real value of the database is in its utility as a behavioral modeler. That is, the designs may subjected to thermal or structural analysis, or behavioral analysis (simulation). The databases that are output from these tools need to be captured and controlled against unwarranted changes, while making them easily accessible to end users.

The system shall not limit the size of any individual file. It must also provide support for compound documents, that is documents represented by multiple files. For example, a MS Word document may have graphics files or spreadsheets as part of the document. Many CAE systems have multiple files and indeed, a separate directory structure for any given design (e.g., CADDS5).

The system must be capable of storing both files and metadata for any given record. The system must support distributed vaults, using replicated or mirrored repositories. The location of information must be transparent to the user, who must simply know that it resides "in the system".

The system must control multiple logical relationships between objects stored in the system such that multiple "views" of the data can be represented in various data structures.

The system must support the creation and use of additional object classes and allow extensions to those classes, either in the addition of attributes, relationships, or methods associated with such objects.

It is desireable to have the system provide for archival of any and all other forms of written or graphical communication, such as email.

9.2.1.1 Search Options

The system must allow users to search the database using any object attribute and value. The system also must support full-text search and retrieval based on content. To support and enhance the performance of the full-text search capability, the system must index objects as they are stored (checked into) the vault.

9.2.1.2 Search Results

The results of searches must be presented as a Web page listing URLs of objects which satisfy the search results, and a subset of the metadata regarding each object. Search results must be displayed within the Web browser, and can be printed using the browsers standard printing utilities. The system must also allow the search results to be saved as an ASCII text file (comma separated or tab-delimited). The objects listed in the search results must have hyperlinks to enable the immediate opening and viewing of the object, and subsequent launching of helper applications if required.

9.2.1.3 Reports

The system must provide utilities for generating status reports as outlined in Section 8.1.2. The system must also provide the user the ability to generate ad-hoc reports based on any piece of metadata (object attribute). The user should have the ability to save and reuse these reports, or make them available to other users. The report generator should require no programming skills on the part of the development team or the user. The user should have the ability to sort the objects in the report based on any attribute as reported in the tabular columns of the report. The reports should be displayed as Web pages (with or without hyperlinked URLs), and the user should have the ability to save the reports as tab-delimited ASCII text files or some other standard format.

9.2.2 DATA ACQUISITION

9.2.1.4 From Legacy Systems

The system must offer a means by which information—CAD data, text files, and metadata about these objects—can be brought into the system from external systems (through copying or links).

It should be possible to create a data domain or subset of the system contents that only contains metadata about objects residing in other repositories. For example, one repository may store and manage component models, related drawings, and documents, while a second repository includes information such as cost, supplier, build schedules, etc., along with pointers or keys to the objects in the first repository. This will support the integration of non-graphic legacy data with the document, model, and drawing management components of the system.

Tools within the system must support the integration of legacy data through various methods, such as APIs, database interface standards (ODBC, JDBC) or screen dump manipulation.

9.2.1.5 To/From Current Systems

The system must be able to exchange information about the objects it controls with other systems. The system must offer options as to how the exchange of data is to be achieved. For example, some systems will require bi-directional exchanges, while others will require uni-directional transfer. Again, some will require real-time interactions, while for others a batch update process will suffice. Some examples of data exchange are outlined in the STEP standards for exchange of PDM data (e.g., AP203).

9.2.1.6 Off-the-Shelf Tools

The system must provide off-the-shelf interfaces and tools for the development of interfaces to common Product Data Management, Local Data Management, ERP, MRP, and other data management systems. In addition, the system must provide "generic" interfaces supporting standards such as ODBC/JDBC to allow the development of interfaces to Government and other highly custom systems.

9.2.3 DATA VAULT AND DOCUMENT MANAGEMENT

Information about engineering products such as models and drawings is called metadata. This information is crucial to enabling the searching for and retrieval of information. The data vault ensures that product data is up-to-date, correct, and protected from accidental or deliberate damage. Check-out and check-in functions provide secure storage, access control, and management of work-in-progress. Through the use of the metadata the data vault may also manage non-electronic information.

9.2.4 WORKFLOW AND PROCESS MANAGEMENT

A PDM system can automate the flow of information throughout its development cycle. The business rules of an organization can be predefined and embedded into the PDM system such that these automated procedures can drive the organization with speedy information flows.

A lifecycle for any object in the system should consist of a set of states such as "in-work", "in-review", "released", "obsolete", etc. The workflow should allow splits in the lifecycle to allow routing along a different path depending on the status of the object in question. For example, a purchase order with a value less than $500 might be routed to a supervisor, while one with a value greater than $10,000 might go to a vice president for approval.

The workflow process should allow customization for the purposes of triggering automated events, such as file conversion, or file transfer, and provide the ability to politely nag people when their deliverables are overdue

9.2.5 PRODUCT STRUCTURE MANAGEMENT

Product structures are enhanced bills-of-materials that may include a complete spacecraft, from the system-level descriptions to the individual nuts and bolts. As-designed and as-built parts lists can easily be generated from the product structure maintained in the PDM system. The system must support multiple views of the product structure and should include the following:

- Lifecycle based (as-designed, as-built, as-maintained, as-flown, etc.)
- Location based (JPL, Goldstone, Madrid, Canberra, Lockheed Martin, ESA, etc.)
- Effectivity based (as of some date, or serial number, or range of dates or numbers)
- Discipline based (mechanical, electrical, software, thermal, mission operations, etc.)
- Any combination of these.

The user must be able to easily filter views within the product tree so as to limit the amount of information displayed. For example, a complete product tree may include software, hardware, and all documentation. But the user should have the capability to limit the components of the tree to only software and documentation, for example.

The system must be able to generate baselines at arbitrary milestone during the life of a project. Baselines contain a snapshot of the significant parts of a product structure in various configurations depending on the lifecycle phase of the project.

The system must support the concept of configuration items. Configuration items are fully tested and certified components that have functional reference designators separate from a part/serial number. They can be purchased parts or built in house, but are nevertheless identified as individual elements within the Work Breakdown Structure or Product Breakdown Structure.

9.2.6 CLASSIFICATION

Metadata can be expanded for any object in the system to include classification criteria. The application of common attributes enables design reuse by simplifying searches based on these common attributes. Classification functions provide efficient mechanisms for accessing standard or similar parts than do catalogs or other manual systems.

9.2.7 PROGRAM MANAGEMENT

Program or project management provides a Work Breakdown Structure (WBS) that can be linked to deliverables that are managed within the PDM system. Where the WBS and the product breakdown structure (PBS) refer the same deliverable (that is, where the output of a task in the WBS is a product), then there can be great efficiencies accrued in the management of overall schedules, budgets, and earned value reporting.

9.2.8 COMMUNICATION AND NOTIFICATION AND SUBSCRIPTION

Communication among design team members is improved through automatic notification of events, such as drawing review process, or drawing release. Email messages are generated or triggered by actions of the users of the PDM system, including changes to status or content of controlled items. Email should also be triggered as a means of reminding that some deliverable is pending, or overdue

9.2.9 DATA TRANSPORT

The PDM system automatically moves data across the network to or from the user. The user does not need to understand the mechanisms of FTP or email attachments. Data transport happens seamlessly and automatically.

9.2.10 DATA TRANSLATION

Data translators can be predefined and invoked through actions of the user. If data is stored in one format but used in another by an application, then the translation of that data into the proper format can be automated. An example would be to store CAD models in a STEP format but translate them into a native CAD format when they are copied out or checked out.

9.2.11 IMAGE SERVICES & VISUALIZATION

Since CAD applications are expensive to buy and to learn, it is not practical to buy a seat for every individual who may wish to view a drawing. The PDM system should manage Web-viewable file formats as well as the native CAD files. This allows relatively inexpensive browsers to be distributed to the user community, and enables the efficient dissemination of information. Because most of mechanical design (and its documentation) are captured in 3-dimensional models, visualization of these models is a high priority requirement. The system should provide a robust visualization tool that allows model parameters such as dimensions or other pertinent metadata to be displayed. Functions such as zoom, pan, rotation, sectioning, dimensioning, etc., should be provided by the system.

The system should also provide users with utilities to automatically convert file formats so as to deliver documents, drawings, and models in a format that they can use with their native tool set.

9.2.12 SYSTEM AND DATABASE ADMINISTRATION

The system and database administrators set up the operational parameters of the PDM system and manage its day to day functions, such as data backup and archiving. Some of these functions relate to application development, such as interface design and functionality. Others include the customization or tailoring of the database schema and functions for the needs of a particular project, such as project member administration, or the identification of new roles and authorization privileges.

9.3 DOCUMENT AND DATA MANAGEMENT (DDM)

Document and Data management consists of four sub-processes: authoring, reviewing, publishing, and archiving.

9.3.1 AUTHORING

The DDM function provides utilities and templates that aid the author in her task. Templates provide embedded guidelines that assist with the identification and codification of best business practices, which have been identified by the organization. This sub-process enables the "re-use" of knowledge regarding preferred and standard authoring practices. The system needs to provide a collaborative work environment that can be shared among team members, utilizing common tools and providing rapid access to their shared work.

9.3.2 REVIEWING

Online reviews are enabled by a workflow engine/process that routes documents and other electronic deliverables (or viewable images of the same) to individuals whose role in the lifecycle of the document is defined in the workflow process. Electronic "distribution lists" are created automatically or semi-automatically and can be modified as needed by the author. Hence, notifications are sent to the reviewers alerting them to their role in the next phase of the lifecycle for the document. Reviews are performed by "redlining" or otherwise marking up a copy of the document and returning that markup to the author or custodian. All of the routing, redlining, and return of the documents should be managed by the review process within the system. Reviews need to be performed on all deliverables for a project, including documents, drawings, mechanical and electronic models (which may be represented in layout drawings or schematic drawings).

9.3.3 PUBLISHING

Publishing is the act of releasing, distributing or otherwise making it known to an audience that a document (or some other deliverable) is available for general use. In an electronic implementation, the placement of the document in an accessible repository, or making it otherwise accessible (by changing its permissions, for example), and with proper notification as to its location, would be an equivalent form of publishing. Part of the DDM functions would provide this publishing activity in an automated manner.

9.3.4 ARCHIVING

At the completion of a project, portions of its documentation will be collected and stored for permanent retention. Copies of such documents, models, software, etc. are "archived" by storing them on some none volatile media that will survive for some number of years, depending on the policies of the project or the Laboratory. The system will provide an archival function that allows such collection and archiving.

10 IMPLEMENTATION STRATEGY

JPL's strategy in moving toward this enterprise system is to pursue a procurement for the components of a system that can be integrated into an operational capability in the least amount of time. These components will include software, hardware, and may include consulting and training as required.

The timetable for this implementation strategy is as follows:

1. Develop and publish an RFI that includes this white paper.
2. From the responses to the RFI, identify the salient points, and review the requirements and implementation strategy. Amend each as necessary.
3. Adopt best business practices where they help achieve JPL's objectives and goals. Use off-the-shelf applications where possible to achieve the best cost/performance ratio.
4. Develop and publish an RFP that includes a complete set of system and subsystem functional requirements that must be met in order to win the contract.
5. Evaluate the RFP responses.
6. Choose two or three finalist to demonstrate their capabilities at JPL with "real" data using real user scenarios.
7. Evaluate the demonstrated capabilities as compared with the system requirements.
8. Award the procurement. Negotiate the contract.
9. Implement the system.

11 SUMMARY

Just as with a word processor or a CAD tool, one size does not fit all, in the case of a product data management system or a document management system. However, it is hoped that this RFI will illuminate the available technologies and solutions available today. It is conceivable that technological enhancements and developments in the database disciplines and with the Internet will continue to change technical capabilities and hence JPL's expectations.

While responses to the RFI necessarily must discuss current capabilities, they may engender subsequent discussions regarding emerging technologies. JPL will adopt an architecture and applications that have been developed with a view toward continued evolution of the Internet and E-commerce (business to business) capabilities. All proposed solutions to this RFI will be evaluated in light of the continued evolution of the Internet as a basis for any viable business model.

Additional information to include:

1. Security (should have a section that covers basic security, which might include):
   - SSL
   - Encryption of passwords and data
   - User roles and permissions
   - Restricting whole classes of information and as well as individual objects

2. Scalability
   - Ability for system administrators and/or project staff to scale the system up or down (i.e., based on capabilities needed by individual project customers).